**Notes for MCA-I (Semester- I)**

**Subject :- Object Oriented Software Engineering**

**(Subject Code:- IT-13)**

**Chapter: 4**] **User Interface Design**

**Introduction :-** The visual part of a computer application or operating system through which a client interacts with a computer or software. It determines how commands are given to the computer or the program and how data is displayed on the screen.

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

* Attractive
* Simple to use
* Responsive in short time
* Clear to understand
* Consistent on all interfacing screens

**Following are the types of design elements:**

**1. Data design elements**

* The data design element produced a model of data that represent a high level of abstraction.
* This model is then more refined into more implementation specific representation which is processed by the computer based system.
* The structure of data is the most important part of the software design.

**2. Architectural design elements**

* The architecture design elements provides us overall view of the system.
* The architectural design element is generally represented as a set of interconnected subsystem that are derived from analysis packages in the requirement model.

**The architecture model is derived from following sources:**

* The information about the application domain to built the software.
* Requirement model elements like data flow diagram or analysis classes, relationship and collaboration between them.
* The architectural style and pattern as per availability.

**3. Interface design elements**

* The interface design elements for software represents the information flow within it and out of the system.
* They communicate between the components defined as part of architecture.

**Following are the important elements of the interface design:**

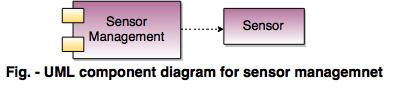
1. The user interface

2. The external interface to the other systems, networks etc.

3. The internal interface between various components.

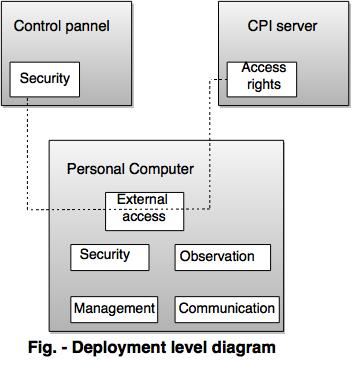
**4. Component level diagram elements**

* The component level design for software is similar to the set of detailed specification of each room in a house.
* The component level design for the software completely describes the internal details of the each software component.
* The processing of data structure occurs in a component and an interface which allows all the component operations.
* In a context of object-oriented software engineering, a component shown in a UML diagram.
* The UML diagram is used to represent the processing logic.



**5. Deployment level design elements**

* The deployment level design element shows the software functionality and subsystem that allocated in the physical computing environment which support the software.
* Following figure shows  three computing environment as shown. These are the personal computer, the CPI server and the Control panel.



* **Types of User Interface**

There are two main types of User Interface:

* Text-Based User Interface or Command Line Interface (CLI)
* Graphical User Interface (GUI)
* **Text-Based User Interface Or CLI:-**

This method relies primarily on the keyboard. A typical example of this is UNIX.

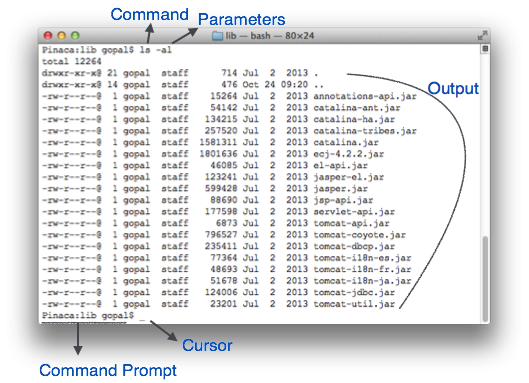
CLI has been a great tool of interaction with computers until the video display monitors came into existence. CLI is first choice of many technical users and programmers. CLI is minimum interface a software can provide to its users.

CLI provides a command prompt, the place where the user types the command and feeds to the system. The user needs to remember the syntax of command and its use. Earlier CLI were not programmed to handle the user errors effectively.

A command is a text-based reference to set of instructions, which are expected to be executed by the system. There are methods like macros, scripts that make it easy for the user to operate.

CLI uses less amount of computer resource as compared to GUI.

**CLI Elements :-**



A text-based command line interface can have the following elements:

* **Command Prompt** - It is text-based notifier that is mostly shows the context in which the user is working. It is generated by the software system.
* **Cursor** - It is a small horizontal line or a vertical bar of the height of line, to represent position of character while typing. Cursor is mostly found in blinking state. It moves as the user writes or deletes something.
* **Command** - A command is an executable instruction. It may have one or more parameters. Output on command execution is shown inline on the screen. When output is produced, command prompt is displayed on the next line.

**Advantages for CLI** :-

* Many and easier to customizations options.
* Typically capable of more important tasks.

**Disadvantages CLI:-**

* Relies heavily on recall rather than recognition.
* Navigation is often more difficult.
* **Graphical User Interface**

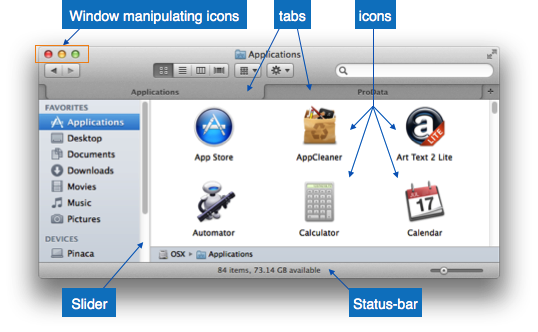
Graphical User Interface provides the user graphical means to interact with the system. GUI can be combination of both hardware and software. Using GUI, user interprets the software.

Typically, GUI is more resource consuming than that of CLI. With advancing technology, the programmers and designers create complex GUI designs that work with more efficiency, accuracy and speed.

**GUI Elements :-**

GUI provides a set of components to interact with software or hardware.

Every graphical component provides a way to work with the system. A GUI system has following elements such as:

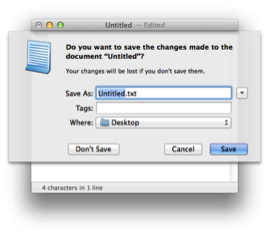


* **Window** - An area where contents of application are displayed. Contents in a window can be displayed in the form of icons or lists, if the window represents file structure. It is easier for a user to navigate in the file system in an exploring window. Windows can be minimized, resized or maximized to the size of screen. They can be moved anywhere on the screen. A window may contain another window of the same application, called child window.
* **Tabs** - If an application allows executing multiple instances of itself, they appear on the screen as separate windows.**Tabbed Document Interface** has come up to open multiple documents in the same window. This interface also helps in viewing preference panel in application. All modern web-browsers use this feature.
* **Menu** - Menu is an array of standard commands, grouped together and placed at a visible place (usually top) inside the application window. The menu can be programmed to appear or hide on mouse clicks.
* **Icon** - An icon is small picture representing an associated application. When these icons are clicked or double clicked, the application window is opened. Icon displays application and programs installed on a system in the form of small pictures.
* **Cursor** - Interacting devices such as mouse, touch pad, digital pen are represented in GUI as cursors. On screen cursor follows the instructions from hardware in almost real-time. Cursors are also named pointers in GUI systems. They are used to select menus, windows and other application features.

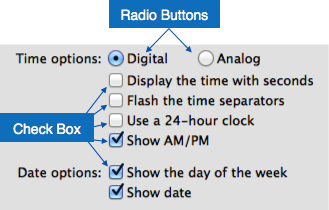
**Application specific GUI components:-**

A GUI of an application contains one or more of the listed GUI elements:

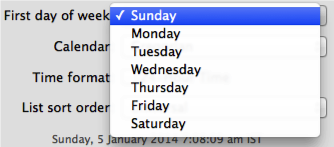
* **Application Window** - Most application windows uses the constructs supplied by operating systems but many use their own customer created windows to contain the contents of application.
* **Dialogue Box**- It is a child window that contains message for the user and request for some action to be taken. For Example: Application generate a dialogue to get confirmation from user to delete a file.



* **Text-Box** - Provides an area for user to type and enter text-based data.
* **Buttons** - They imitate real life buttons and are used to submit inputs to the software.



* **Radio-button** - Displays available options for selection. Only one can be selected among all offered.
* **Check-box** - Functions similar to list-box. When an option is selected, the box is marked as checked. Multiple options represented by check boxes can be selected.
* **List-box**- Provides list of available items for selection. More than one item can be selected.



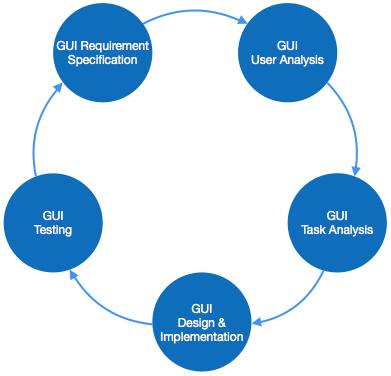
**Other impressive GUI components are:**

* Sliders
* Combo-box
* Data-grid
* Drop-down list

**User Interface Design Activities :-**

There are a number of activities performed for designing user interface. The process of GUI design and implementation is alike SDLC. Any model can be used for GUI implementation among Waterfall, Iterative or Spiral Model.

A model used for GUI design and development should fulfill these GUI specific steps.



* **GUI Requirement Gathering** - The designers may like to have list of all functional and non-functional requirements of GUI. This can be taken from user and their existing software solution.
* **User Analysis** - The designer studies who is going to use the software GUI. The target audience matters as the design details change according to the knowledge and competency level of the user. If user is technical savvy, advanced and complex GUI can be incorporated. For a novice user, more information is included on how-to of software.
* **Task Analysis** - Designers have to analyze what task is to be done by the software solution. Here in GUI, it does not matter how it will be done. Tasks can be represented in hierarchical manner taking one major task and dividing it further into smaller sub-tasks. Tasks provide goals for GUI presentation. Flow of information among sub-tasks determines the flow of GUI contents in the software.
* **GUI Design & implementation** - Designers after having information about requirements, tasks and user environment, design the GUI and implements into code and embed the GUI with working or dummy software in the background. It is then self-tested by the developers.
* **Testing** - GUI testing can be done in various ways. Organization can have in-house inspection, direct involvement of users and release of beta version are few of them. Testing may include usability, compatibility, user acceptance etc.

**GUI Implementation Tools**

There are several tools available using which the designers can create entire GUI on a mouse click. Some tools can be embedded into the software environment (IDE).

GUI implementation tools provide powerful array of GUI controls. For software customization, designers can change the code accordingly.

There are different segments of GUI tools according to their different use and platform.

Example

Mobile GUI, Computer GUI, Touch-Screen GUI etc. Here is a list of few tools which come handy to build GUI:

* FLUID
* AppInventor (Android)
* LucidChart
* Wavemaker
* Visual Studio
* **4.2 User Interface Golden rules :-**

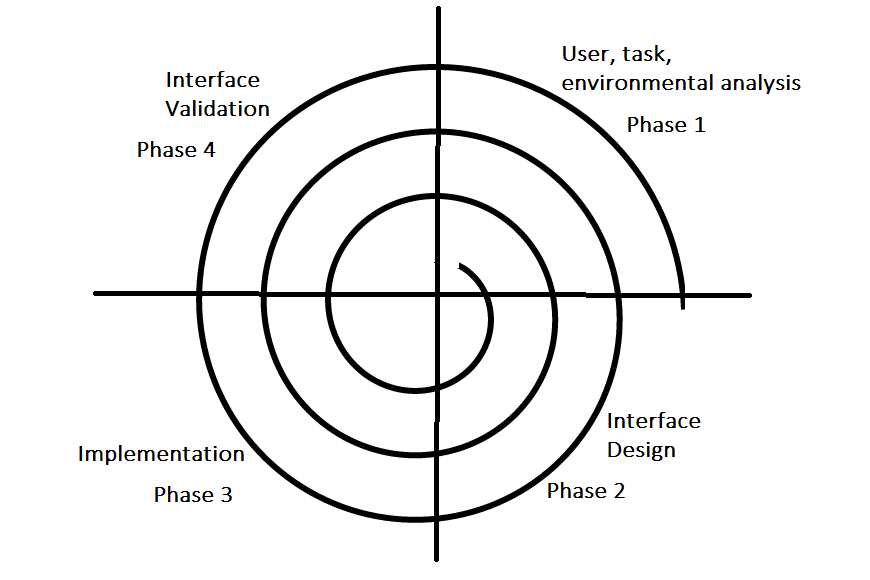
The following rules are mentioned to be the golden rules for GUI design, described by Shneiderman and Plaisant in their book (Designing the User Interface).

* **Strive for consistency** - Consistent sequences of actions should be required in similar situations. Identical terminology should be used in prompts, menus, and help screens. Consistent commands should be employed throughout.
* **Enable frequent users to use short-cuts** - The user’s desire to reduce the number of interactions increases with the frequency of use. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.
* **Offer informative feedback** - For every operator action, there should be some system feedback. For frequent and minor actions, the response must be modest, while for infrequent and major actions, the response must be more substantial.
* **Design dialog to yield closure** - Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and this indicates that the way ahead is clear to prepare for the next group of actions.
* **Offer simple error handling** - As much as possible, design the system so the user will not make a serious error. If an error is made, the system should be able to detect it and offer simple, comprehensible mechanisms for handling the error.
* **Permit easy reversal of actions** - This feature relieves anxiety, since the user knows that errors can be undone. Easy reversal of actions encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.
* **Support internal locus of control** - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
* **Reduce short-term memory load** - The limitation of human information processing in short-term memory requires the displays to be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

**GUI Characteristics**

|  |  |
| --- | --- |
| **Characteristics** | **Descriptions** |
| Windows | Multiple windows allow different information to be displayed simultaneously on the user's screen. |
| Icons | Icons different types of information. On some systems, icons represent files. On other icons describes processes. |
| Menus | Commands are selected from a menu rather than typed in a command language. |
| Pointing | A pointing device such as a mouse is used for selecting choices from a menu or indicating items of interests in a window. |
| Graphics | Graphics elements can be mixed with text or the same display. |

**User Interface Design Process:-**



The analysis and design process of a user interface is iterative and can be represented by a spiral model. The analysis and design process of user interface consists of four framework activities.

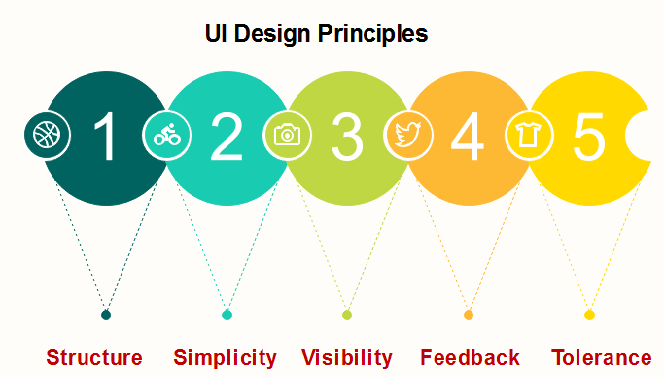
1. **User, task, environmental analysis, and modeling:** Initially, the focus is based on the profile of users who will interact with the system, i.e. understanding, skill and knowledge, type of user, etc, based on the user’s profile users are made into categories. From each category requirements are gathered. Based on the requirements developer understand how to develop the interface. Once all the requirements are gathered a detailed analysis is conducted. In the analysis part, the tasks that the user performs to establish the goals of the system are identified, described and elaborated. The analysis of the user environment focuses on the physical work environment. Among the questions to be asked are:
   * Where will the interface be located physically?
   * Will the user be sitting, standing, or performing other tasks unrelated to the interface?
   * Does the interface hardware accommodate space, light, or noise constraints?
   * Are there special human factors considerations driven by environmental factors?
2. **Interface Design:** The goal of this phase is to define the set of interface objects and actions i.e. Control mechanisms that enable the user to perform desired tasks. Indicate how these control mechanisms affect the system. Specify the action sequence of tasks and subtasks, also called a user scenario. Indicate the state of the system when the user performs a particular task. Always follow the three golden rules stated by Theo Mandel. Design issues such as response time, command and action structure, error handling, and help facilities are considered as the design model is refined. This phase serves as the foundation for the implementation phase.
3. **Interface construction and implementation:** The implementation activity begins with the creation of prototype (model) that enables usage scenarios to be evaluated. As iterative design process continues a User Interface toolkit that allows the creation of windows, menus, device interaction, error messages, commands, and many other elements of an interactive environment can be used for completing the construction of an interface.
4. **Interface Validation:** This phase focuses on testing the interface. The interface should be in such a way that it should be able to perform tasks correctly and it should be able to handle a variety of tasks. It should achieve all the user’s requirements. It should be easy to use and easy to learn. Users should accept the interface as a useful one in their work.

**Advantages GUI**:-

* Less expert knowledge is required to use it.
* Easier to Navigate and can look through folders quickly in a guess and check manner.
* The user may switch quickly from one task to another and can interact with several different applications.

**Disadvantages GUI** :-

* Typically decreased options.
* Usually less customizable. Not easy to use one button for tons of different variations.
* **UI Design Principles:-**



**Structure:** Design should organize the user interface purposefully, in the meaningful and usual based on precise, consistent models that are apparent and recognizable to users, putting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another. The structure principle is concerned with overall user interface architecture.

**Simplicity:** The design should make the simple, common task easy, communicating clearly and directly in the user's language, and providing good shortcuts that are meaningfully related to longer procedures.

**Visibility:** The design should make all required options and materials for a given function visible without distracting the user with extraneous or redundant data.

**Feedback:** The design should keep users informed of actions or interpretation, changes of state or condition, and bugs or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users.

**Tolerance:** The design should be flexible and tolerant, decreasing the cost of errors and misuse by allowing undoing and redoing while also preventing bugs wherever possible by tolerating varied inputs and sequences and by interpreting all reasonable actions.

* **4.3 features of GUI :-**

**Window:-**

This is the element that displays the information on the screen. It is very easy to manipulate a window. It can be opened or closed with the click of an icon. Moreover, it can be moved to any area by dragging it around. In a multitasking environment, multiple windows can be open at the same time, all of them performing different tasks.

There are multiple types of windows in a graphical user interface, such as container window, browser window, text terminal window, child window, message window etc.

A window is an area on the screen that displays information, with its contents being displayed independently from the rest of the screen. An example of a window is what appears on the screen when the "My Documents" [icon](https://en.wikipedia.org/wiki/Icon_(computing)) is clicked in the [Windows Operating System](https://en.wikipedia.org/wiki/Microsoft_Windows). It is easy for a user to manipulate a window: it can be shown and hidden by clicking on an icon or [application](https://en.wikipedia.org/wiki/Application_software), and it can be moved to any area by dragging it (that is, by clicking in a certain area of the window – usually the title bar along the top – and keeping the pointing device's button pressed, then moving the pointing device). A window can be placed in front or behind another window, its size can be adjusted, and scrollbars can be used to navigate the sections within it. Multiple windows can also be open at one time, in which case each window can display a different application or file – this is very useful when working in a multitasking environment. The system memory is the only limitation to the number of windows that can be open at once. There are also many types of specialized windows.

* A **container window** encloses other windows or controls. When it is moved or resized, the enclosed items move, resize, reorient, or are clipped by the container window.
* A [**browser window**](https://en.wikipedia.org/wiki/Web_browser) allows the user to view and navigate through a collection of items, such as files or [web pages](https://en.wikipedia.org/wiki/Web_page). [Web browsers](https://en.wikipedia.org/wiki/Web_browser) are an example of these types of windows.
* [**Text terminal**](https://en.wikipedia.org/wiki/Computer_terminal#Text_terminals) windows present a character-based, command-driven [text user interfaces](https://en.wikipedia.org/wiki/Text_user_interface) within the overall graphical interface. [MS-DOS](https://en.wikipedia.org/wiki/MS-DOS) and [UNIX](https://en.wikipedia.org/wiki/Unix) consoles are examples of these types of windows. Terminal windows often conform to the hotkey and display conventions of CRT-based terminals that predate GUIs, such as the VT-100.
* A **child window** opens automatically or as a result of a user activity in a parent window. [Pop-up windows](https://en.wikipedia.org/wiki/Pop-up_ad) on the [Internet](https://en.wikipedia.org/wiki/Internet) can be child windows.
* A **message window**, or [**dialog box**](https://en.wikipedia.org/wiki/Dialog_box), is a type of child window. These are usually small and basic windows that are opened by a program to display information to the user and/or get information from the user. They almost always have one or more buttons, which allow the user to dismiss the dialog with an affirmative, negative, or neutral response.

**Menu :-**

A menu contains a list a choices and it allows users to select one from them. A menu bar is displayed horizontally across the screen such as pull down menu. When any option is clicked in this menu, then the pull down menu appears.

Another type of menu is the context menu that appears only when the user performs a specific action. An example of this is pressing the right mouse button. When this is done, a menu will appear under the cursor.

Menus allow the user to execute commands by selecting from a list of choices. Options are selected with a mouse or other pointing device within a GUI. A keyboard may also be used. Menus are convenient because they show what commands are available within the [software](https://en.wikipedia.org/wiki/Software). This limits the amount of documentation the user reads to understand the software.

* A [**menu bar**](https://en.wikipedia.org/wiki/Menu_bar) is displayed horizontally across the top of the screen and/or along the tops of some or all windows. A pull-down menu is commonly associated with this menu type. When a user clicks on a menu option the pull-down menu will appear.
* A [**menu**](https://en.wikipedia.org/wiki/Menu_(computing)) has a visible title within the menu bar. Its contents are only revealed when the user selects it with a pointer. The user is then able to select the items within the pull-down menu. When the user clicks elsewhere the content of the menu will disappear.
* A [**context menu**](https://en.wikipedia.org/wiki/Context_menu) is invisible until the user performs a specific mouse action, like pressing the right mouse button. When the software-specific mouse action occurs the menu will appear under the cursor.
* [**Menu extras**](https://en.wikipedia.org/wiki/Menu_extra) are individual items within or at the side of a menu.

**Icons:-**

Files, programs, web pages etc. can be represented using a small picture in a graphical user interface. This picture is known as an icon. Using an icon is a fast way to open documents, run programs etc. because clicking on them yields instant access.

An [icon](https://en.wikipedia.org/wiki/Icon_(computing)) is a small picture that represents objects such as a file, program, web page, or command. They are a quick way to execute commands, open documents, and run programs. Icons are also very useful when searching for an object in a browser list, because in many operating systems all documents using the same extension will have the same icon.

**Panels** :-

are collections of related faces used for specific parts of a user interface. The faces within a **panel** are created during the layout stage of processing where the **GUI** language (a dialect) is interpreted and its styles create actual faces.

Basically, panels provide a way to:

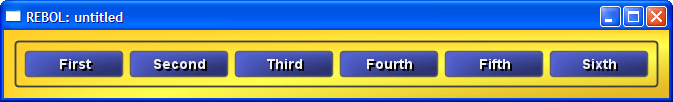
* Group a number of faces together,
* Arrange faces into a desired layout,
* Display a 2D layer, with background or other effects,
* Update and resize those faces when events occur.

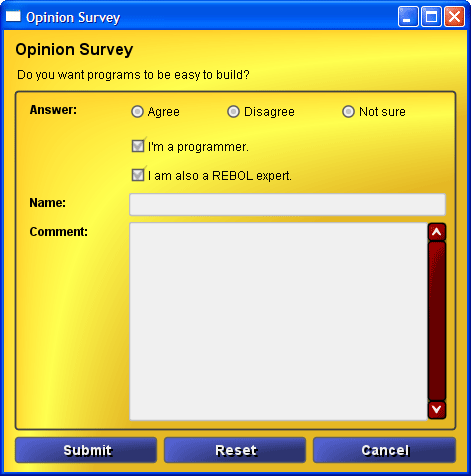
Every [window](http://www.rebol.com/r3/docs/view/windows.html) of the GUI display is a panel, and each contains a number of sub-panels to further divide its area into sections. And, a panel is itself a face, and a GUI is created from one or more layers of panels, each of which holds faces of its own.

The faces that compose a panel are defined using the GUI language, a special dialect of REBOL. This approach saves time and effort compared to function-based or tag-based languages.

**Using Panels**

Panels are fundamental to the GUI. When you open a new window like the one defined below, you are creating a panel





* **Error message**

It is information displayed when an unforeseen problem occurs, usually on a [computer](https://en.wikipedia.org/wiki/Computer_science) or other device. On modern operating systems with [graphical](https://en.wikipedia.org/wiki/Graphical), error messages are often displayed using dialog boxes. Error messages are used when user intervention is required, to indicate that a desired operation has failed, or to relay important warnings (such as warning a computer user that they are almost out of [hard disk](https://en.wikipedia.org/wiki/Hard_disk_drive) space). Error messages are seen widely throughout computing, and are part of every [operating system](https://en.wikipedia.org/wiki/Operating_system) or [computer hardware](https://en.wikipedia.org/wiki/Personal_computer_hardware) device. Proper design of error messages is an important topic in [usability](https://en.wikipedia.org/wiki/Usability) and other fields of [human–computer interaction](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction).

**The following error messages are commonly seen by modern computer users:**

**Access denied**

This error occurs if the user doesn't have privileges to a file, or if it has been [locked](https://en.wikipedia.org/wiki/File_locking) by some program or user.

**Device not ready**

This error most often occurs when there is no [floppy disk](https://en.wikipedia.org/wiki/Floppy_disk) (or a bad disk) in the disk drive and the system tries to perform tasks involving this disk.

**File not found**

The file concerned may have been damaged, moved, deleted, or a [bug](https://en.wikipedia.org/wiki/Software_bug) may have caused the error. Alternatively, the file simply might not exist, or the user has mistyped its name. More frequent on [command line interfaces](https://en.wikipedia.org/wiki/Command_line_interface) than on  [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface) where files are presented [iconically](https://en.wikipedia.org/wiki/Computer_icon) and users do not type file names.

**Low Disk Space**

This error occurs when the hard drive is (nearly) full. To fix this, the user should close some programs (to free [swap file](https://en.wikipedia.org/wiki/Virtual_memory) usage) and delete some files (normally temporary files, or other files after they have been backed up), or get a bigger hard drive.

**Out of memory**

This error occurs when the system has run out of memory or tries to load a file too large to store in [RAM](https://en.wikipedia.org/wiki/Random-access_memory). The fix is to close some programs or install more memory.

**[program name] has stopped working.**

This message is displayed by Microsoft [Windows 10](https://en.wikipedia.org/wiki/Windows_10) when a program causes a [general protection fault](https://en.wikipedia.org/wiki/General_protection_fault) or invalid [page fault](https://en.wikipedia.org/wiki/Page_fault).

